Problema 2

A versão modificado tem como objetivo trazer uma melhor establidade numérica ao método de Gram-Schmidt. A ideia é que, ao invés de calcular a projeção de um vetor sobre os outros já ortogonalizados, calculamos a projeção do vetor mais atualizados sobre os já ortogonalizados. Isso evita a acumulação de erros numéricos.

Implementação

qr_GSM.m

```
% Entradas:
%   A - matriz (m x n)
% Saídas:
%   Q = matriz (m x n) ortogonal
%   R = matriz (n x n) triangular superior
function [Q,R] = qr_GSM(A)
  [m, n] = size(A);

  % Inicializa matrizes
  Q = zeros(m,n);
  R = zeros(n);

  for j = 1 : n
    v = A(:,j); % j-ésima coluna de A

    % Obtém, por Gram-Schmidt, v o j-ésimo vetor de uma base ortogonal
    for i = 1 : j-1
      R(i,j) = dot(Q(:,i), v); % Passa a usar o vetor mais atualizado
      v = v - R(i,j) * Q(:,i);
    end

    R(j,j) = norm(v,2);
    Q(:,j) = v / R(j,j); % j-ésimo vetor de uma base ortonormal
  end
end
```

Testes

Para a matriz A, essa modificação obteve os mesmos resultaos que a anterior. Portanto, não há necessidade expor os resultados.

Para a matriz B, uma matriz má condicionada, a modificação alterou minimamente uma entrada da matriz R. Quanto à matriz Q, notamos que a ultima colunas dessa nova Q é diferente da anterior.

```
>> QCb(:,4)' % Quarta coluna da matriz Q obtida com qr_GSC
    0.32233    0.40291    0.64466   -0.56408

>> [QMb, RMb] = qr_GSM(B);

>> QMb(:,4)'
    0.94679    0.063119    0.25248   -0.18936
```

Associada a essa mudança, vemos uma melhor ortogonalidade da matriz Q utilizando o método de Gram-Schmidt modificado. Contudo, devido à dimensão da matriz esse ganho é mínimo.

Considerando matrizes mágicas de ordem par (caracterizadas por serem muito mal condicionadas), fica claro que esse algoritmo produz uma matriz Q muito mais próxima da ortogonalidade.

```
>> [Q ~] = qr_GS(magic(20));
>> [QM ~] = qr_GSM(magic(20));
>> norm(Q'*Q - eye(20))
    16.989
>> norm(QM'*QM - eye(20))
    0.99972

>> [Q ~] = qr_GS(magic(100));
>> [QM ~] = qr_GSM(magic(100));
>> norm(Q'*Q - eye(100))
    97
>> norm(QM'*QM - eye(100))
    0.99995
```

Resta testar a matriz C. Agora, com essa modificação, deixamos de obter uma fatoração eficaz. Ambas as matrizes possus entradas NaN (não faz sentido verificar ortogonalidade ou acurácia). Isso se deve ao fato de C possuir colunas LD (linearmente dependentes). Logo essa modificação do método restringe a função a matrizes LI (linearmente independentes).

```
>> [QMc, RMc] = qr_GSM(C)
QMc =
     0.5547    0.83205    NaN    NaN
    0.83205    -0.5547    NaN    NaN

RMc
    3.6056     1.3868    2.7735    3.8829
         0    0.27735    0.5547    2.2188
         0          0         0       NaN
         0          0         0       NaN
```